A Software Solution For Making Digital Systems For Data Sorting

Pavle Savković[1], Branislav Atlagić[2], Nikola Vranić [3], Vladimir Marinković[4]

Abstract - **The paper presents an approach to the design of digital data-sorting system in parameterized HDL source code, and software for its customization and generation. As a case study of presented concept, it is described an optimal solution for data sorting system that receives input data in serial way.**

Keywords - **data sorting, parallel digital systems, FPGA**

I. INTRODUCTION

An effective data sorting is an obligatory problem in every complex system, and it plays an important role in many disciplines like: automatic control in industry, robotics, artificial intelligence, genetic algorithms, medicine etc. Need for efficient sorting algorithms is evident from great number of published works on this subject. [1]. Most of those algorithms are intended to work on different processor types. Recently, due to increase in equivalent gate count, development of dedicated primitives used for image processing, as well as lowering power consumption, FPGA (*Field-Programmable Gate Array*) devices slowly starts to replace some segments or even a whole traditional image processing systems. Beside dedicated DSP (*Digital Signal Processing*) primitives, potential for parallel computing is greatest advantage of FPGA-based devices. This improvements lead to development of new algorithms for data sorting and tailoring existing ones.

One of the classifications of the sorting algorithms, realized in the digital systems, is based on the ways of data reception:

- Serial – the data arrives in the system one after another
- Completely parallel – all data arrive in the system simultaneously
- Partially parallel – the data arrives in the system partially in series and parallel way.

To make the development of digital systems for data sorting easier dedicated software environment is developed.

[1]Pavle Savković, is with Faculty of Technical Sciences at University of Novi Sad, Fruškogorska 18, Novi Sad, 21000, Serbia, E-mail: Pavle.Savkovic@krt.neobee.net

[2]Branislav Atlagić, is with Faculty of Technical Sciences at University of Novi Sad, Fruškogorska 18, Novi Sad, 21000, Serbia, E-mail: Branislav.Atlagic@krt.neobee.net

[3]Nikola Vranić, is with Faculty of Technical Sciences at University of Novi Sad, Fruškogorska 18, Novi Sad, 21000, Serbia, E-mail: Nikola.Vranic@krt.neobee.net

[4]Vladimir Marinković, is with Faculty of Technical Sciences at University of Novi Sad, Fruškogorska 18, Novi Sad, 21000, Serbia, E-mail: Vladimir.Marinkovic@krt.neobee.net

This software incorporates following modules:

- SortGen – a software tool module for calculating the parameters and synthesis the digital systems for data sorting,
- SortProbe – a software tool module for test & verification of the developed data sorting systems and
- SortLib – Verilog library of parameterized source code for definition, synthesis and verification (testbench development) of the data sorting system

Using the generated testbench, SortProbe tool facilitate testing of the sorting system by sending test data and receiving results, via serial interface to the target hardware platform. This encircles the development and testing process of the digital system for data sorting.

The simplest system for realization of all sorting systems is the system where data comes serially. That was the reason why this type of data reception was realized first. The chapter two contains description of the algorithms and digital system implementation. The chapter three contains the detailed explanation of the SortGen software tool. The forth chapter contains description of the SortProbe tool, and in the end, the chapter five contains the results/comments for few different FPGA based system realization.

II. DESCRIPTION OF DIGITAL SYSTEM FOR DATA SORTING

In this project, we consider the case of sorting in which the members of array are not known at the beginning of the sorting, but they are coming in regular time period. Besides this constraint, the system should enable that:

- data array must be sorted in any time,
- the maximal period for sorting entire data array after arrival of new data item must be less than one system cycle, and
- the source code must be organized in the way that enables parameterization.

These constrains will be meet only if the data array is sorted before the moment of new data arrival (i.e. less than period of the one system cycle). The most important parts of the system for sorting are shown in the Figure 1. Because of the large number of modules and the connections between them, only the most important parts and their connections are shown. The registers for storing inputs, from R0 to Rn, and one of the n pairs of the multiplexer-comparator, which are the basic elements of system for sorting the array of the elements length n, are shown. Output from multiplexer 0 represents minimal, and output form multiplexer n-1 represents maximal value in data array. The algorithm is based on idea to compare new

input item simultaneously to all array data, and than do the calculation of new values for multiplexer's selection.

The data items are stored in one of the registers immediately after their arrival. When system starts to work,

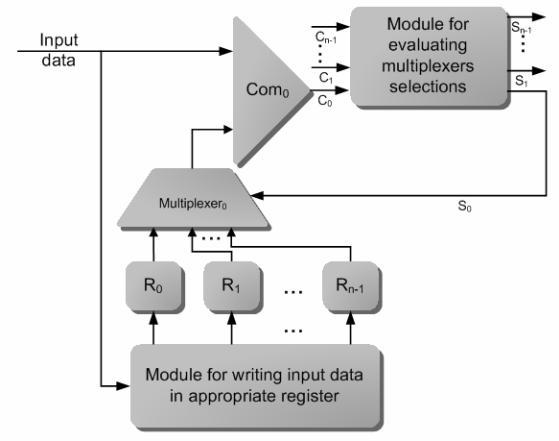

Figure 1. Block diagram of key parts of system for sorting data

the first arrived item is stored in the register R0, the next one in R1, until the nth item that is stored in the register Rn-1. The next one, n-th data item is again stored in the register R0, and so on, following the circular scheme for writing in the registers. In which one of the registers the data is stored depends on value of a dedicated counter with modulo n. Receiving data in cyclic manner is chosen to minimize number of registers. Algorithm for data sorting does not require any substitution of register's contents. The number of registers is equal to the number of members in data array that should be sorted. To avoid changing values between registers multiplexers n-to-1 are used. The number of multiplexers is equal to the number of the registers for data array. The output from multiplexer is connected to the comparator, which compares the value from the selected register and the current input item. All of the registers are brought at the multiplexers inputs. The selections come from the module for calculating the new selection values.

The fact that the new data is inserted in previously sorted array makes the calculation of the new selection values easier[2]. The outputs from comparators follow certain pattern. This will be explained in the next example, where the array of 9 elements is to be sorted. The output from comparators is shown in the Figure 2. The comparators 0 to 5 show that the new value is higher than the value that is brought on those comparators, while the comparators from 6 to 8 show that the new value is smaller. Because of that the

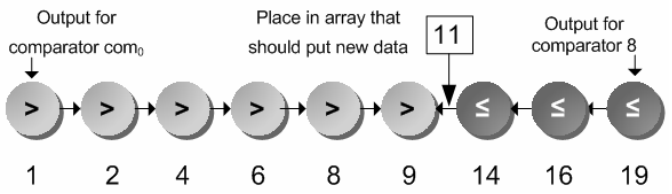

Figure 2. Example of comparators output

conclusion is that the new value should be stored at the 6th place, counting from 0. This example proves that the logic for the updating multiplexer selections is simple and it could be parameterized easily.

The module for the evaluating multiplexer selection contains logic for determining the new values for multiplexer selections, which are to be valid during the next cycle period, and the dedicated registers for their storing until this cycle begins. This solution fulfills all demanded requirements, i.e. sorting is done within one system cycle, and data array remains sorted all the time.

III. Description of SortGen Software

Possibility to parameterize the digital system leads to development of SortGen software. Software, according to user's requirements, calculates system parameters and generates custom source code using library of parameterized source code. The first version of the SortGen tool was console application and it was used for initial testing. The second version introduces a graphic user interface (GUI). Figure 3 shows the main window SortGen tools.

To generate source code, it is necessary to enter array length, the data width, sorting order (increasing or decreasing order) and finally which data will be from the sorting system will be forwarded to its output. To output from the system for

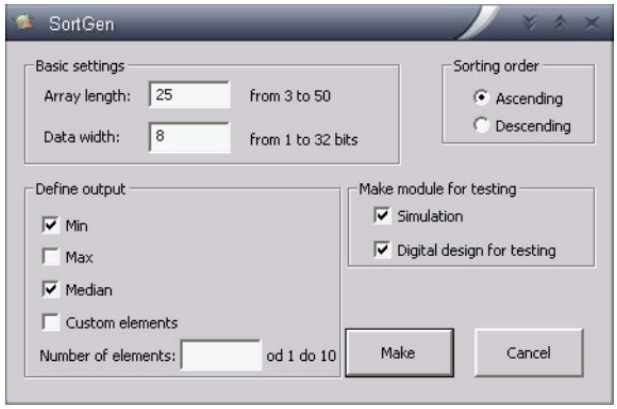

Figure 3. Main window SortGen tools

data sorting can be forwarded, individually or in any combination, minimum, maximum and median value of sorted array. Also user can define any other position in array to be forwarded to output. This option is added in order to completey customize the system to the user's needs.

When the user enters all the necessary information with confirmation on button "Make", tool will create 15 files with the source code in the same directory where the tool is settled. All the source code files are fully accessible to user.

A. The Description Of The Digital System For Data Sorting

The software can optionally make testbench and complete digital system for the testing the system for sorting on FPGA. The Figure 4 presents the testbench organization.

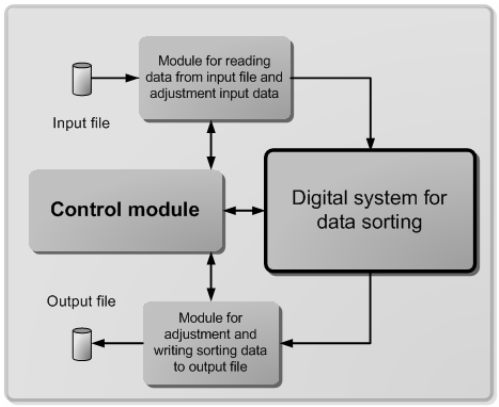

Figure 4. Testbench organization

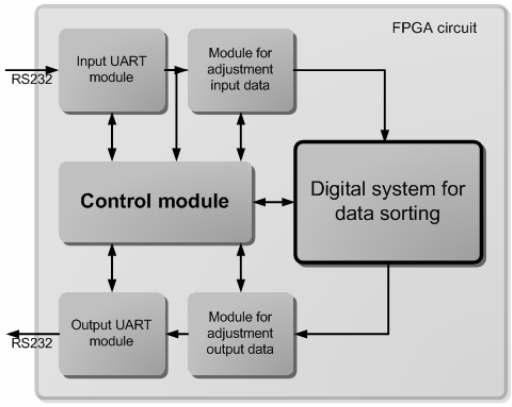

Figure 5. The organization of the digital system for testing digital system of sorting the facts

The testbench is composed of four modules. The task of the module for reading and adapting the values from input file is to read one by one data from the input file and to send it to the control module and digital system under test. The input file should be formatted in such a way to contain in every line just one value, which will be forwarded to the system under test and control module, without any other symbols, expect decimal numbers (hexadecimal and binary numbers are not supported). The duration of simulation is determined by the number of values in the input file. If the value, which goes out of the tested range, is brought in the file, or if it is brought into any sign, which is not the number, there will be indicated a mistake and the simulation will be stopped. The control module is responsible for the control of the simulation's progress and checking of the output values during the simulation. In order to speed up the testing and to shorten the time of simulation, the control module models digital system for sorting, using the constructions of Verilog language, which could not be synthesized. It compares the outputs of the digital system with results, which are got in the model of simulation's system and if any differences are detected, it will issue the warning to the user, who can in that moment stop the simulation, without need to wait its ending. Although the possibility of simulation's check during its operation is incorporated, the results of the sorting are written in the output file. The special module is responsible for formatting the data and their writing in the output file. The format of the output file is the same as the format of the input file.

B. The Description Of The System For Testing Digital Sorting System On FPGA

Beside the testbench, the SortGen software is able to generate complete digital system for testing digital system on FPGA[3]. The organization of the testing system is given in Figure 5.

The SortProbe software (it will be explained in details in the next chapter) sends via serial interface commands and data to the digital system realized in FPGA device. When system starts the operation, SortProbe first establishes the connection with system under test and then it sends the commands for setting system parameters. All of that is followed by the sending data and commands and reconvening results. The receiving UART module (Universal Asynchronous Receiver / Transmitter) is responsible for the serial receiving data and commands from SortProbe software.

The width of the word, is sent or received by UART interface, is predefined to 8 bits. The most significant bit in each word sent over serial interface indicates what is sent, data or command. The controls relate to adapting the parameters in the system and in the communication between the digital system and SortProbe software. The task of the receiving UART module is to interpret received message, to send commands to the command module and the data to the system for sorting and to the control module. The module for adapting input data takes over from the receiving UART module certain number of words and forms them in the data, which are intended to the digital system for sorting. For example, if the width of the data, which are sorted by the system, is 24 bits, it is needed to take over 4 words from the receiving UART module and to make of them one 24-bits data word and to send it to the system for sorting. It is needed four 8-bits words for sending 24-bits of the data, because every package contains one bit of the indication, which shows if the sent package, is data or command.

The results of the digital system for sorting are in the same way formed for sending. The forming of the data performs the module for the adapting the results, before it sends them to the Delivery UART module, which sends them to SortProbe software via UART interface. The control module controls the system and supervises the progress of the receiving and sending the data. If the control module finds out that there is a possibility of making the mistake in the data flow, it could stop every individual module and/or send message to SortProbe software via UART interface. The stopping of the digital system is performed synchronously in order not to exert influence on its function and not to imperil the integrity of the data flow in the system.

IV. THE DESCRIPTION OF THE PROGRAM SORTPROBE SOFTWARE

The main task of the SortProbe software is to enable sending the data and commands to the digital system for testing the system for data sorting and to enable the

supervision of results. The Figure 6 presents the main window of SortProbe software.

Before the start of the operation it is necessary to enter the following information: a array length, which should be sorted, the width of the data in bits, if the system is tested by the data from the user's file or random array of numbers should be generated, then the length of the tested sequence. After the entering all of the necessary data, by confirmation on the button "Run test", begins the testing of the digital system for data sorting.

The instantaneous status of the testing is marked by three indicators: red one, yellow one and green one. If there is not any mistake or warning during the testing – the green indicator is active. If there is a warning – the yellow indicator

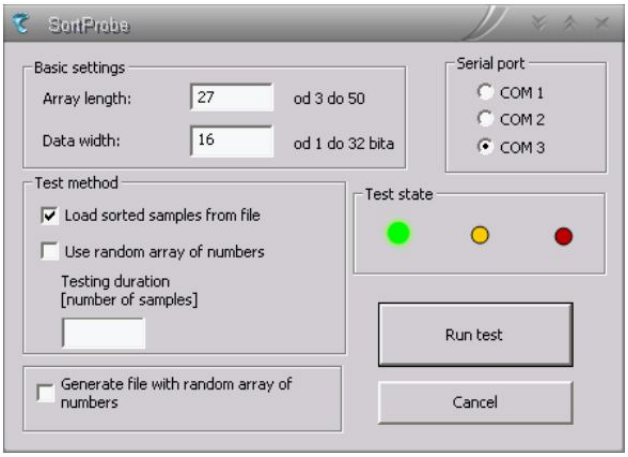

Figure: 6. The main window of SortProbe software

is active (for example, the input buffer overflow), and if there is a mistake – the red indicator is active. The receiving results and the messages are put in two separated files, which can be later reviewed by user.

V. The Results Of The Realization Of The Digital Sorting System

The testing is performed on digital sorting system for sorting a array of 25 bytes. The digital system and testbench for it was generated using SortGen software. The results of the simulation were correct. Next step was making the digital system for the testing on the FPGA. The made system is tested on the platform for fast development of the digital systems, which is based on Xilinx's FPGA Spartan 3 device. The testing is done by the help of the SortProbe software. For the testing the special test files are made. The results of the testing are checked using SortProbe software and the SciLab software package for verification. Both software gave the same results.

In order to get information about the quantity of the resources, which the digital sorting system takes, it is made a synthesis of the systems for few of the most used FPGA devices of Xilinx production, whose software for synthesis is free. The results are given in the Table 1.

TABLE I
THE OCCUPATION OF THE RESOURCE IN THE DIFFERENT FPGA CIRCUITS

			Array length		
			9	25	35
FPGA circuit	SPARTAN 3 XSC200 FG900 -5	LUTs	704 (1%)	5284 (12%)	8578 (20%)
		ROM	6	22	30
		MHz	85	65.9	62.1
	VIRTEX 2P VP50 FF1152 -6	LUTs	745 (1%)	5284 (10%)	8641 (17%)
		ROM	4	22	26
		MHz	102	72.3	68.3
	VIRTEX 4 VLX80 FF1148 -11	LUTs	713 (<1%)	5555 (7%)	8564 (11%)
		ROM	6	22	31
		MHz	132	97.8	92.1
	VIRTEX 5 LX220 FF1760 -2	LUTs	510 (<1%)	3414 (2%)	5648 (4%)
		ROM	4	18	27
		MHz	141	108.4	98.8

The Table 1 shows how many resources has the system for the date sorting, whose length is 9, 25 and 35 members, when the synthesis for different FPGA devices is done. The number of ROM primitives (Read-only memory) and logical program cells describes, how much the digital system is demanded about the resource questions. The frequency expressed in MHz presents the maximal speed, on which the system can operate without any mistake.

VI. Conclusion

In this work is described an environment for the design digital systems for data sorting and algorithm for data sorting for the system, in which data arrive serially. All of the goals, which are placed in front of the algorithm before the projecting, are satisfied. The example of the realization of the digital system for data sorting of the 25 members shows that the system is absolutely functional and that it make easier and speeds up design of the complex digital systems. The development of extension, for described system, for parallel reception and parallel sorting based on sorting networks is in progress.

References

[1] Leighton, Frank Thomson, "Introduction to Parallel Algorithms and Architectures-array, tress, hypercubes", San Mateo Morgan Kaufmann Publishers, Inc. 1992

[2] Donald E. Knuth, "The art of computer programming, volume 3 : sorting and searching" Addison Wesley Longman Publishing Co. ,Inc., Redwood city, Ca,1998

[3] Hahanov,V Kaminska,M. "Testability Analysis of Digital Design Verification", Baltic Electronic Conference, 2006